CS3230 Tutorial 1

Asymptotic Analysis

Join our telegram group!
https://t.me/+8N_QNiKQQ-xIMDY1

(AY 25/26 Semester 2)

January 19, 2026

(Prepared by Benson)

Reminder for Attendance Taking: Write the last **two characters** of your Matric No. on the whiteboard. (e.g. if your Matric No. is `A0123456X`, write `6X`.)

Contents

Admin Stuff

1. *(Only for this tutorial group)* Complete (and submit) the pre-survey form (9/14) and join the telegram group (10/14).
2. Join the discord server for the course (N/A, can't track).

Introduction

YEUNG Man Tsung (Benson)

- Third time teaching CS3230!
  - Also teaching CS2109S and CS3211 this sem.

- Year 4 Computer Science

- **Email**: mtyeung@u.nus.edu
- **Discord**: @mtyeung
- **Telegram**: @mtyeung

- Consultations: By appointment (I'm generally free on Monday afternoons)

About CS3230

CS1231(S) Discrete Structures

$+$

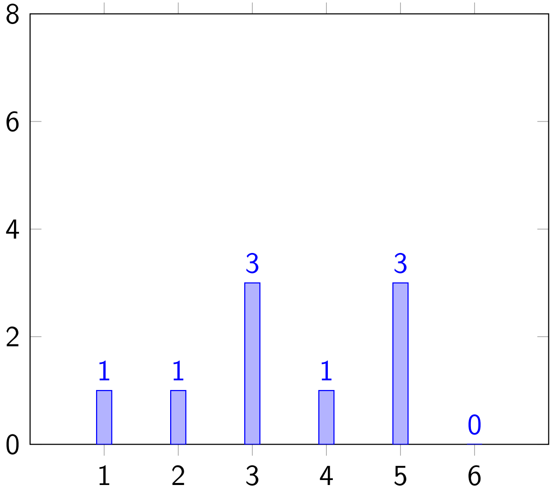
CS2040(C/S)
Data Structures and Algorithms

$\Rightarrow$

CS3230
Design and Analysis of Algorithms

About CS3230

How confident are you in your mathematical proving competency?

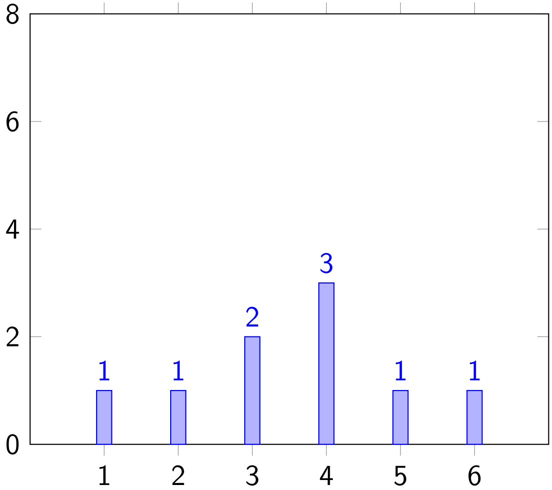


This course will be (mathematically) rigorous. Be prepared.

If you totally forgot how to prove things, read How to Create a Proof (by Allan Yashinski) before the semester gets busy.

About CS3230

How confident are you in your data structures and algorithms competency?



There will be *some* algorithm design involved (Divide & Conquer, Greedy, DP).

About CS3230 Tutorials

Format:

- We will discuss 4-6 problems per tutorial.
- It's OK if you do not attempt them beforehand, but please at least watch the lecture before you come.
- Attendance + participation marks make up **5%** of your final grade.

Attendance Marks:

- 1 mark for each tutorial. (free marks for public holidays)
- Write the **last two characters** of your Matric No. on the whiteboard (e.g. if your Matric No. is `A0123456X`, write `6X`). Each student can only write one entry (monitored by everyone else).
- I will take a picture of the board after the tutorial.

About CS3230 Tutorials

Participation Marks:

- Each student will be assigned 2 tutorials which you are expected to ~~present a solution~~ answer some (quick) questions.
- 5 marks awarded for each reasonable attempt.
- Check the assignment on my website, no swaps allowed.
  - Limited make-up opportunities available for the last 2 tutorials.

Bonus Questions *(optional)*:

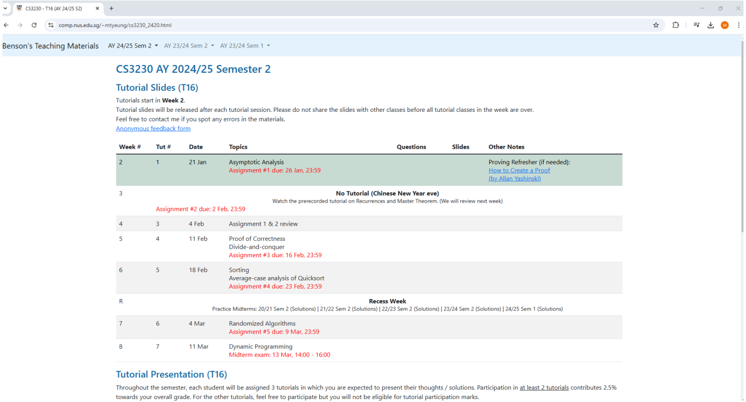
- If you feel the tutorials are too easy, you may attempt the bonus question *during* the tutorial. Send me your solutions *within* the tutorial hour.
- Claim one participation for every two bonus questions (fully) solved.

Total Marks: 1×12 (Attendance) $+ 5 \times 2$ (Participation), Capped at 20 marks

About CS3230 Tutorials

Resources: https://www.comp.nus.edu.sg/~mtyeung/cs3230_2520.html

- Tutorial slides and recording (no guarantees, best effort).
- Other supplementary materials.
- Assignment for tutorial participation.



CS3230 - T16 (AY 24/25 S2)

comp.nus.edu.sg/~mtyeung/cs3230_2420.html

Benson's Teaching Materials AY 24/25 Sem 2 AY 23/24 Sem 2 AY 23/24 Sem 1

CS3230 AY 2024/25 Semester 2

Tutorial Slides (T16)

Tutorials start in **Week 2**.
Tutorial slides will be released after each tutorial session. Please do not share the slides with other classes before all tutorial classes in the week are over.
Feel free to contact me if you spot any errors in the materials.
Anonymous feedback form

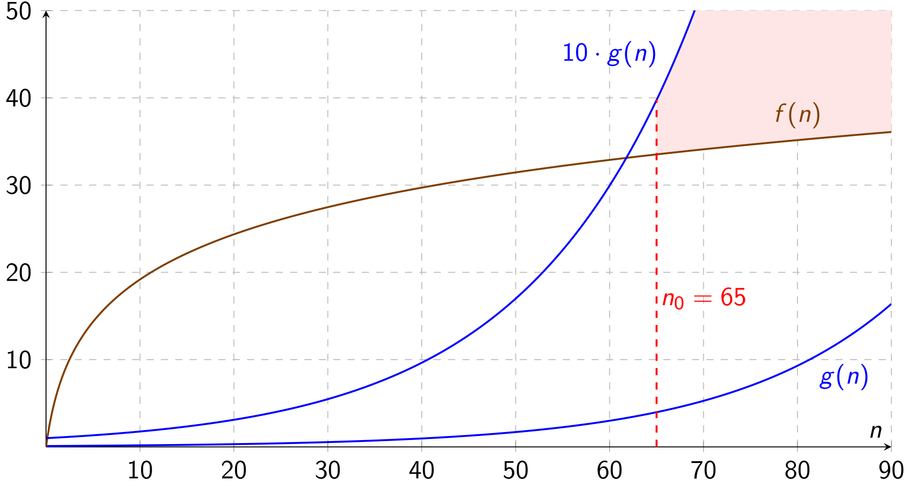
Week #	Tut #	Date	Topics	Questions	Slides	Other Notes
2	1	21 Jan	Asymptotic Analysis Assignment #1 due: 26 Jan, 23:59			Proving Refresher (if needed): How to Create a Proof (by Allan Yashinski)
3			**No Tutorial (Chinese New Year eve)** Watch the prerecorded tutorial on Recurrences and Master Theorem. (We will review next week) Assignment #2 due: 2 Feb, 23:59			
4	3	4 Feb	Assignment 1 & 2 review			
5	4	11 Feb	Proof of Correctness Divide-and-conquer Assignment #3 due: 16 Feb, 23:59			
6	5	18 Feb	Sorting Average-case analysis of Quicksort Assignment #4 due: 23 Feb, 23:59			
R			**Recess Week** Practice Midterms: 20/21 Sem 2 (Solutions) | 21/22 Sem 2 (Solutions) | 22/23 Sem 2 (Solutions) | 23/24 Sem 2 (Solutions) | 24/25 Sem 1 (Solutions)			
7	6	4 Mar	Randomized Algorithms Assignment #5 due: 9 Mar, 23:59			
8	7	11 Mar	Dynamic Programming Midterm exam: 13 Mar, 14:00 - 16:00			

Tutorial Presentation (T16)

Throughout the semester, each student will be assigned 3 tutorials in which you are expected to present their thoughts / solutions. Participation in at least 2 tutorials contributes 2.5% towards your overall grade. For the other tutorials, feel free to participate but you will not be eligible for tutorial participation marks.

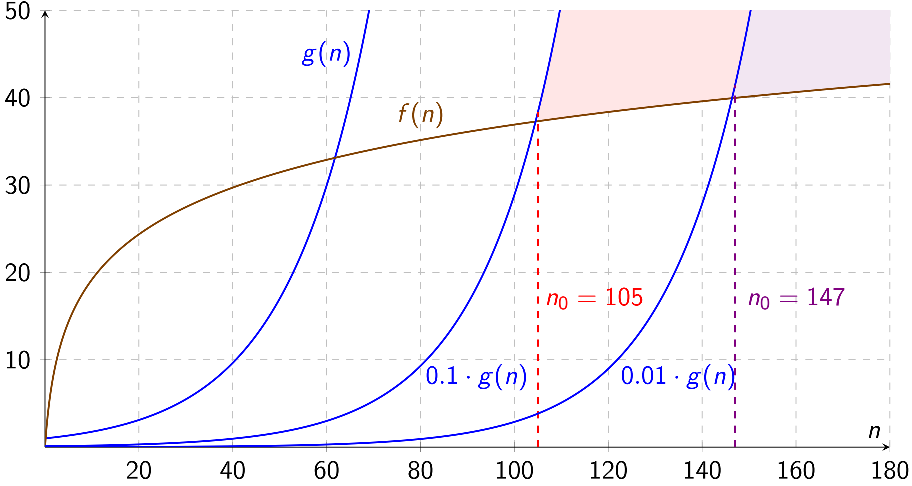
Recap: Asymptotic Notations

$$f(n) = O(g(n)) \Leftrightarrow \exists c\ \exists n_0\ \forall n \geq n_0,\ f(n) \leq c \cdot g(n)$$



Recap: Asymptotic Notations

$$f(n) = o(g(n)) \Leftrightarrow \forall c\ \exists n_0\ \forall n \geq n_0,\ f(n) < c \cdot g(n)$$



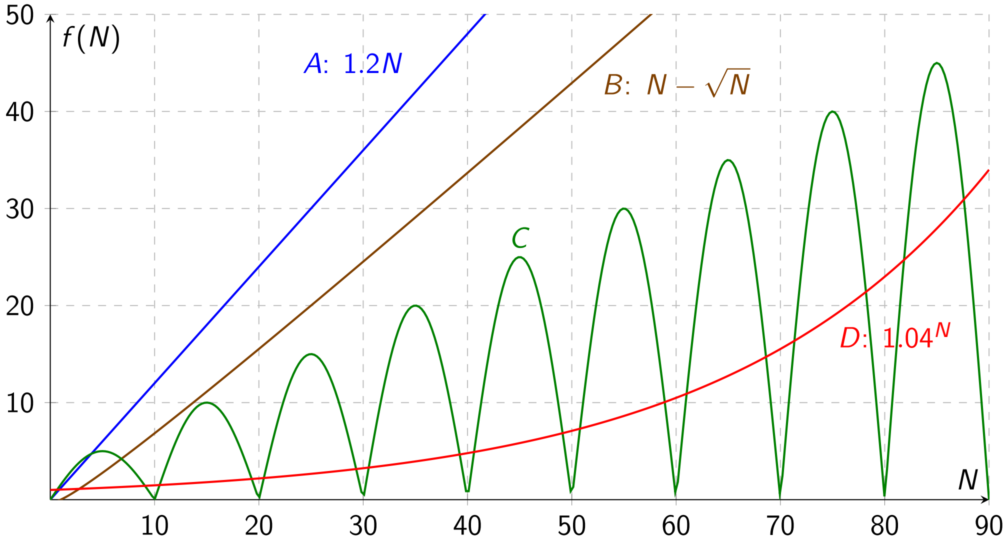
Recap: Asymptotic Notations

- Big-O Notation (**Upper Bound** $\leq$):
  $f(n) = O(g(n))$ iff $\exists c$ $\exists n_0\ \forall n \geq n_0,$ $f(n) \leq c \cdot g(n)$
- Big-Omega Notation (**Lower Bound** $\geq$):
  $f(n) = \Omega(g(n))$ iff $\exists c$ $\exists n_0\ \forall n \geq n_0,$ $f(n) \geq c \cdot g(n)$
- Big-Theta Notation (**Tight Bound**):
  $f(n) = \Theta(g(n))$ iff $\exists c_1, c_2$ $\exists n_0\ \forall n \geq n_0,$ $c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$
- Little o Notation (**Strict Upper Bound** $<$):
  $f(n) = o(g(n))$ iff $\forall c$ $\exists n_0\ \forall n \geq n_0,$ $f(n) < c \cdot g(n)$
- Little Omega Notation (**Strict Lower Bound** $>$):
  $f(n) = \omega(g(n))$ iff $\forall c$ $\exists n_0\ \forall n \geq n_0,$ $f(n) > c \cdot g(n)$

(Note: We assume $c, c_1, c_2, n_0 > 0$ and that all functions are asymptotically nonnegative to be useful)

Recap: Asymptotic Notations

Poll: Consider the following plot. Which of the functions is/are in $o(N)$?



Recap: Asymptotic Notations

Solution:

A. FALSE. This is $\Theta(N)$, obviously.

B. FALSE. This is $\Theta(N)$. Note that $\frac{1}{2}N \leq N - \sqrt{N} \leq N$ for $N > 4$.

C. FALSE. If we consider the peaks $N = 5, 15, 25, \ldots$, the function grows linearly. Hence it is not $o(N)$. It is not $\Omega(N)$ either, since it reaches 0 periodically.

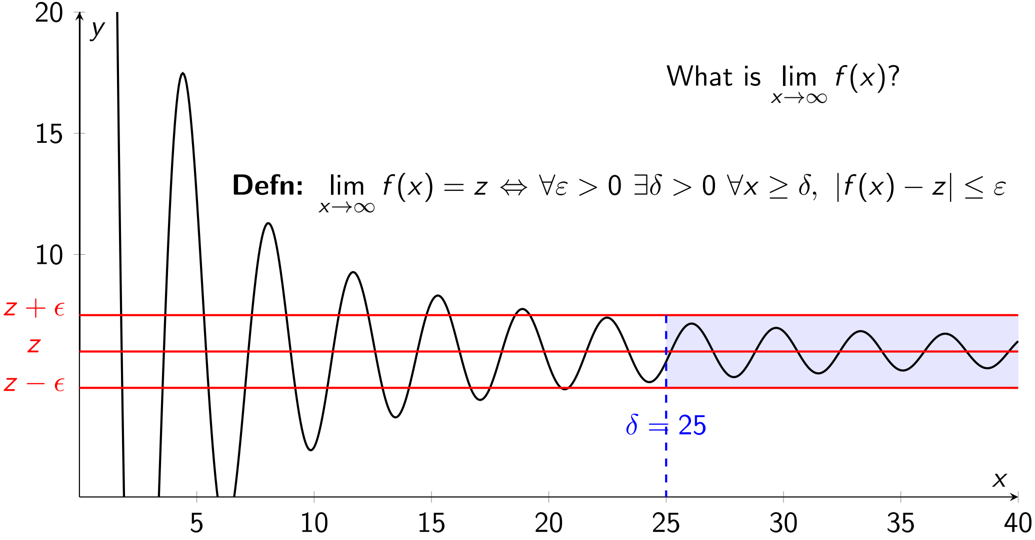
D. FALSE. $1.04^N > N$ for $N > 123$.

Q1. Taking Limits

Assume $f(n), g(n) > 0$. Show

- $\lim_{n\to\infty}\left(\frac{f(n)}{g(n)}\right) = 0 \quad \Rightarrow f(n) = o(g(n))$
- $\lim_{n\to\infty}\left(\frac{f(n)}{g(n)}\right) < \infty \quad \Rightarrow f(n) = O(g(n))$
- $0 < \lim_{n\to\infty}\left(\frac{f(n)}{g(n)}\right) < \infty \quad \Rightarrow f(n) = \Theta(g(n))$
- $\lim_{n\to\infty}\left(\frac{f(n)}{g(n)}\right) > 0 \quad \Rightarrow f(n) = \Omega(g(n))$
- $\lim_{n\to\infty}\left(\frac{f(n)}{g(n)}\right) = \infty \quad \Rightarrow f(n) = \omega(g(n))$

Q1. Taking Limits



What is $\lim_{x \to \infty} f(x)$?

Defn: $\lim_{x \to \infty} f(x) = z \Leftrightarrow \forall \varepsilon > 0\ \exists \delta > 0\ \forall x \geq \delta,\ |f(x) - z| \leq \varepsilon$

Q1. Taking Limits

How does **limits** relate to **asymptotic notations**?

- $\lim_{n\to\infty}\left(\frac{f(n)}{g(n)}\right) = z$: $\forall \varepsilon$ $\exists \delta$ $\forall x \geq \delta,$ $\left|\frac{f(x)}{g(x)} - z\right| \leq \varepsilon$
- $f(n) = O(g(n))$: $\exists c$ $\exists n_0$ $\forall n \geq n_0,$ $f(n) \leq c \cdot g(n)$

$\frac{f(x)}{g(x)} - z \leq \varepsilon$
$f(x) \leq (z+\varepsilon)g(x)$

$\frac{f(x)}{g(x)} - z \geq -\varepsilon$
$f(x) \geq (z-\varepsilon)g(x)$

Aim: Show $\lim_{n\to\infty}\left(\frac{f(n)}{g(n)}\right) < \infty \Rightarrow f(n) = O(g(n))$.

Q1. Taking Limits

Show $\lim_{n\to\infty}\left(\frac{f(n)}{g(n)}\right) < \infty \Rightarrow f(n) = O(g(n))$.

- By definition of limit, $\lim_{n\to\infty}\left(\frac{f(n)}{g(n)}\right) = z$ means:

  For all $\varepsilon > 0$, there exists a $n_0 > 0$, such that for all $n \geq n_0$, $\frac{f(n)}{g(n)} \leq z + \varepsilon$.
- Take $\varepsilon = 0.1$, there exists a $n_0 > 0$, such that for all $n \geq n_0$, $\frac{f(n)}{g(n)} \leq z + 0.1$.
- For $c = z + 0.1$, there exists $n_0 > 0$ such that for all $n \geq n_0$, $f(n) \leq c \cdot g(n)$.
- Hence $f(n) = O(g(n))$.

Q1. Taking Limits

- $\lim_{n\to\infty}\left(\frac{f(n)}{g(n)}\right) = z$: $\forall \varepsilon$ $\exists \delta$ $\forall x \geq \delta$ $\left|\frac{f(x)}{g(x)} - z\right| \leq \varepsilon$

$\frac{f(x)}{g(x)} - z \leq \varepsilon$
$f(x) \leq (z+\varepsilon)g(x)$

$\frac{f(x)}{g(x)} - z \geq -\varepsilon$
$f(x) \geq (z-\varepsilon)g(x)$

Intuition:

- $f(n) = o(g(n))$: $\forall c$ $\exists n_0$ $\forall n \geq n_0$ $f(n) < c \cdot g(n)$
  - When $z = 0$, we have $\forall \varepsilon$ $f(x) \leq \varepsilon \cdot g(x)$.
- $f(n) = \Theta(g(n))$: $\exists c_1, c_2$ $\exists n_0$ $\forall n \geq n_0$ $c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$
  - When $0 < z < \infty$, we have $(z - \varepsilon) \cdot g(x) \leq f(x) \leq (z + \varepsilon) \cdot g(x)$.
  - Take a **sufficiently small** ε (e.g. $z/2$).
- $f(n) = \Omega(g(n))$: $\exists c$ $\exists n_0$ $\forall n \geq n_0$ $f(n) \geq c \cdot g(n)$
  - When $z > 0$, we have $f(x) \geq (z - \varepsilon)g(x)$.
  - Take a **sufficiently small** ε (e.g. $z/2$).
- $f(n) = \omega(g(n))$: $\forall c$ $\exists n_0$ $\forall n \geq n_0$ $f(n) > c \cdot g(n)$
  - When $z = \infty$, then $\forall \varepsilon$ $f(x) \geq \varepsilon \cdot g(x)$.

Note: For o and ω you can choose n_0 appropriately to remove equality.

Q1. Taking Limits

These laws only applies in one direction, i.e. if the limit exists, then the value can be determined. However, the converse does not necessarily apply, i.e. if $f(n) = O(g(n))$, then the limit $< \infty$ is is not always true.

Q2. Some Properties

Assume $f(n), g(n) > 0$. Show

- Reflexivity: For O, Ω, Θ,
  $f(n) = O(f(n))$.
- Transitivity: For all five: $O, \Omega, \Theta, o, \omega$,
  $f(n) = O(g(n)) \wedge g(n) = O(h(n)) \Rightarrow f(n) = O(h(n))$.
- Symmetry: $f(n) = \Theta(g(n))$ iff $g(n) = \Theta(f(n))$.
- Complementarity: $f(n) = O(g(n))$ iff $g(n) = \Omega(f(n))$
  $f(n) = o(g(n))$ iff $g(n) = \omega(f(n))$

Q2. Some Properties

Complementarity: $f(n) = O(g(n))$ iff $g(n) = \Omega(f(n))$

- ($\Rightarrow$) Suppose $f(n) = O(g(n))$.
- Then there exist $c, n_0 > 0$, such that $f(n) \leq c \cdot g(n)$ for $n \geq n_0$.
- Hence, for $c' = \boxed{\frac{1}{c}}$ and $n_0' = \boxed{n_0}$, $c' \cdot f(n) \leq g(n)$ for all $n \geq n_0'$.
- Thus, $g(n) = \Omega(f(n))$.

- ($\Leftarrow$) Suppose $f(n) = \Omega(g(n))$.
- Then there exist $c, n_0 > 0$, such that $c \cdot f(n) \leq g(n)$ for $n \geq n_0$.
- Hence, for $c' = \boxed{\frac{1}{c}}$ and $n_0' = \boxed{n_0}$, $f(n) \leq c' \cdot g(n)$ for all $n \geq n_0'$.
- Thus, $g(n) = O(f(n))$.

$f(n) = O(g(n))$	iff	$\exists c$	$\exists n_0 \ \forall n \geq n_0,$	$f(n) \leq c \cdot g(n)$
$f(n) = \Omega(g(n))$	iff	$\exists c$	$\exists n_0 \ \forall n \geq n_0,$	$f(n) \geq c \cdot g(n)$
$f(n) = \Theta(g(n))$	iff	$\exists c_1, c_2$	$\exists n_0 \ \forall n \geq n_0,$	$c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$
$f(n) = o(g(n))$	iff	$\forall c$	$\exists n_0 \ \forall n \geq n_0,$	$f(n) < c \cdot g(n)$
$f(n) = \omega(g(n))$	iff	$\forall c$	$\exists n_0 \ \forall n \geq n_0,$	$c \cdot g(n) < f(n)$

Q3. MRQ

Which of the following statements are true (several may be true):

(a) $3^{n+1} = O(3^n)$

(b) $4^n = O(2^n)$

(c) $2^{\lfloor \log n \rfloor} = \Theta(n)$

(d) For a constant i, $a > 0$ $\quad (n + a)^i = \Theta(n^i)$

Q3. MRQ

(a) $3^{n+1} = O(3^n)$ TRUE

Take $c = 3,\ n_0 = 1,\ \forall n \geq n_0,\ 3^{n+1} \leq 3 \cdot 3^n$.

$f(n) = O(g(n))$	iff	$\exists c$	$\exists n_0\ \forall n \geq n_0,$	$f(n) \leq c \cdot g(n)$
$f(n) = \Omega(g(n))$	iff	$\exists c$	$\exists n_0\ \forall n \geq n_0,$	$f(n) \geq c \cdot g(n)$
$f(n) = \Theta(g(n))$	iff	$\exists c_1, c_2$	$\exists n_0\ \forall n \geq n_0,$	$c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$
$f(n) = o(g(n))$	iff	$\forall c$	$\exists n_0\ \forall n \geq n_0,$	$f(n) < c \cdot g(n)$
$f(n) = \omega(g(n))$	iff	$\forall c$	$\exists n_0\ \forall n \geq n_0,$	$c \cdot g(n) < f(n)$

Q3. MRQ

CS1231/S Refresher: To disprove a statement, **show the negation** of the statement.

(b) $4^n = O(2^n)$ FALSE

We want to show that $\forall c\ \forall n_0\ \exists n \geq n_0\ f(n) > c \cdot g(n)$.

- Take arbitrary $c, n_0 > 0$.
- Take $n = \max(\log c, n_0) + 1$.
- Then $4^n = 2^n \cdot 2^n > c \cdot 2^n$.

We need the following to hold for n:

- $2^n > c$, i.e. $n > \log c$.
- $n \geq n_0$

$f(n) = O(g(n))$	iff	$\exists c$	$\exists n_0\ \forall n \geq n_0,$	$f(n) \leq c \cdot g(n)$
$f(n) = \Omega(g(n))$	iff	$\exists c$	$\exists n_0\ \forall n \geq n_0,$	$f(n) \geq c \cdot g(n)$
$f(n) = \Theta(g(n))$	iff	$\exists c_1, c_2$	$\exists n_0\ \forall n \geq n_0,$	$c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$
$f(n) = o(g(n))$	iff	$\forall c$	$\exists n_0\ \forall n \geq n_0,$	$f(n) < c \cdot g(n)$
$f(n) = \omega(g(n))$	iff	$\forall c$	$\exists n_0\ \forall n \geq n_0,$	$c \cdot g(n) < f(n)$

Q3. MRQ

(c) $2^{\lfloor \log n \rfloor} = \Theta(n)$ TRUE

- Take $c_1 = \frac{1}{2}, c_2 = 1, n_0 = 1$.
- For all $n \geq n_0$, $\frac{1}{2}n = \frac{1}{2}2^{\log n} = 2^{\log n - 1} \leq 2^{\lfloor \log n \rfloor} \leq 2^{\log n} = n$.

$f(n) = O(g(n))$	iff	$\exists c$	$\exists n_0 \ \forall n \geq n_0,$	$f(n) \leq c \cdot g(n)$
$f(n) = \Omega(g(n))$	iff	$\exists c$	$\exists n_0 \ \forall n \geq n_0,$	$f(n) \geq c \cdot g(n)$
$f(n) = \Theta(g(n))$	iff	$\exists c_1, c_2$	$\exists n_0 \ \forall n \geq n_0,$	$c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$
$f(n) = o(g(n))$	iff	$\forall c$	$\exists n_0 \ \forall n \geq n_0,$	$f(n) < c \cdot g(n)$
$f(n) = \omega(g(n))$	iff	$\forall c$	$\exists n_0 \ \forall n \geq n_0,$	$c \cdot g(n) < f(n)$

Q3. MRQ

(d) For a constant i, $a > 0$ $\quad (n+a)^i = \Theta(n^i)$ TRUE

- Take $c_1 = 1, c_2 = 2^i, n_0 = a$.
- For all $n \geq n_0$, $n^i \leq (n+a)^i \leq (2n)^i \leq 2^i \cdot n^i$.

$f(n) = O(g(n))$	iff	$\exists c$	$\exists n_0 \ \forall n \geq n_0,$	$f(n) \leq c \cdot g(n)$
$f(n) = \Omega(g(n))$	iff	$\exists c$	$\exists n_0 \ \forall n \geq n_0,$	$f(n) \geq c \cdot g(n)$
$f(n) = \Theta(g(n))$	iff	$\exists c_1, c_2$	$\exists n_0 \ \forall n \geq n_0,$	$c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$
$f(n) = o(g(n))$	iff	$\forall c$	$\exists n_0 \ \forall n \geq n_0,$	$f(n) < c \cdot g(n)$
$f(n) = \omega(g(n))$	iff	$\forall c$	$\exists n_0 \ \forall n \geq n_0,$	$c \cdot g(n) < f(n)$

Q4. $2^{\log_2 n}$

Which of the following statements are true (several may be true):
$2^{\log_2 n}$ is

(a) $O(n)$ TRUE

(b) $\Omega(n)$ TRUE

(c) $\Theta(\sqrt{n})$ FALSE

(d) $\omega(n)$ FALSE

$2^{\log_2 n} = n$

Q4b. $2^{\log_4 n}$

Which of the following statements are true (several may be true):
$2^{\log_4 n}$ is

(a) $O(n)$

(b) $\Omega(n)$

(c) $\Theta(\sqrt{n})$

(d) $\omega(n)$

Q4b. $2^{\log_4 n}$

Which of the following statements are true (several may be true):
$2^{\log_4 n}$ is

(a) $O(n)$ TRUE

(b) $\Omega(n)$ FALSE

(c) $\Theta(\sqrt{n})$ TRUE

(d) $\omega(n)$ FALSE

$$2^{\log_4 n} = 2^{\frac{\log_2 n}{\log_2 4}} = \left(2^{\log_2 n}\right)^{\frac{1}{\log_2 4}} = n^{\frac{1}{\log_2 4}} = n^{1/2} = \sqrt{n}.$$

Q4b. $2^{\log_4 n}$

Note that for different bases, i.e. $\log_2 n$ and $\log_3 n$, they have the same order of growth, i.e $\log_2 n = \Theta(\log_3 n)$. This is due to the fact that $\log_2 n = (\log_2 3) \cdot \log_3 n$.

It's common in asymptotic analysis that we *ignore* the constants as an easily simplification step. However, it must be noted that constants **must not be ignored** if they are in the exponents, i.e. 2^n and 2^{2n} have different orders of growth, as we have seen.

Q5. Ranking Functions

Rank the following functions by their order of growth (if two of them have same order of growth, mention so).

- $f_1(n) = \log n$
- $f_2(n) = n!$
- $f_3(n) = 2^n + n$
- $f_4(n) = n^{2.3} + 16n + \log n$
- $f_5(n) = \log(n^2)$
- $f_6(n) = \ln(n^{2n})$

Q5. Ranking Functions

Rank the following functions by their order of growth (if two of them have same order of growth, mention so).

- $f_1(n) = \log n$
- $f_2(n) = n!$ $\quad \log n! = \Theta(n \log n)$
- $f_3(n) = 2^n + n$ $\quad \log 2^n = n \log 2$
- $f_4(n) = n^{2.3} + 16n + \log n$
- $f_5(n) = \log(n^2)$ $= 2 \log n$
- $f_6(n) = \ln(n^{2n})$ $= 2n \ln n$

$$\begin{aligned}
\log n! &= \log 1 + \log 2 + \ldots + \log n \\
&\leq \log n + \log n + \ldots + \log n \\
&= n \log n \\
\log n! &= \log 1 + \log 2 + \ldots + \log(n-1) + \log n \\
&= \log(1 \times n) + \log(2 \times (n-1)) + \ldots \\
&\geq \log n + \log n + \log n + \ldots \\
&= \frac{n}{2} \log n
\end{aligned}$$

Logarithmic: $f_1(n) = f_5(n)$. Polynomial: $f_6(n) < f_4(n)$.
Exponential: $f_2(n)$ and $f_3(n)$ (what is their relative order?)
Answer: $f_1(n) = f_5(n) < f_6(n) < f_4(n) < f_3(n) < f_2(n)$.

We don't have time for this...

Read (or print) Asymptotic_Analysis-Useful_Facts.pdf. **IMPORTANT!!**
(FAQ: This version is more comprehensive than the one on Canvas. For my tutorial group, you may quote results from either of them in Assignment 1.)

Feedback

Anonymous Feedback (throughout the semester):

https://forms.gle/TwHsFBZy7pL6fstx6

The link to the tutorial slides will be posted in telegram.
Please do not share the slides with other classes before all tutorial classes in the week are over.

Bonus. Ranking Functions 2.0

Note: Bonus questions are entirely optional. They can get very challenging and/or fall outside the course curriculum. Interested individuals may attempt.

(AY 19/20 Sem 2 Midterm) Rank the following functions in increasing order of growth. Assume: All logarithms are in base 2.

- $g_1(n) = \sum_{i=1}^{\log n - 1} \log \log \frac{n}{2^i}$
- $g_2(n) = \sum_{i=1}^{n-2} \log \log(n - i)$
- $g_3(n) = (\log n)!$
- $g_4(n) = 2^{\log \log \log n}$
- $g_5(n) = 10^{\log((\log \log n)!) / \log \log n}$
- $g_6(n) = \log((\log n)!)$
- $g_7(n) = n^3$
- $g_8(n) = 2^n$

Bonus. Ranking Functions 2.0

Extra Slide

$$
\begin{aligned}
g_1(n) &= \sum_{i=1}^{\log n - 1} \log\log \frac{n}{2^i} \\
&= \log\log n + \log\log \frac{n}{2} + \log\log \frac{n}{4} + \cdots \\
&= \log(\log n) + \log(\log n - 1) + \log(\log n - 2) + \cdots \\
&= \log((\log n)!) \\
&= \Theta(\log n \log\log n) && \blacktriangleleft\ \log(n!) = \Theta(n \log n)
\end{aligned}
$$

$$
\begin{aligned}
g_2(n) &= \sum_{i=1}^{n-2} \log\log(n - i) \\
&= \log\log n + \log\log(n-1) + \log\log(n-2) + \cdots \\
&= \Theta(n \log\log n) && \blacktriangleleft\ \text{use a similar argument}
\end{aligned}
$$

Refer to how we show $\log n! = \Theta(n \log n)$ in the earlier slides.

Bonus. Ranking Functions 2.0

Extra Slide

$$
\begin{aligned}
g_3(n) &= (\log n)! \\
&\approx \sqrt{2\pi \log n}\left(\frac{\log n}{e}\right)^{\log n} \\
&= \Theta\left((\log n)^{\log n + 1/2}\right)
\end{aligned}
$$

◀ Stirling's approximation

$$
g_4(n) = 2^{\log \log \log n} = 2^{\log(\log \log n)} = \Theta(\log \log n)
$$

Bonus. Ranking Functions 2.0

Extra Slide

$$
\begin{aligned}
g_5(n) &= 10^{\log((\log\log n)!)/\log\log n} \\
&= \left(2^{\log((\log\log n)!)/\log\log n}\right)^{\log 10} \\
&\leq \left(2^{c\log\log n\log\log\log n/\log\log n}\right)^{\log 10} && \blacktriangleleft\ \log(n!) = \Theta(n\log n) \\
&\leq \left(2^{c\log\log\log n}\right)^{\log 10} && \blacktriangleleft\ \text{division} \\
&= \Theta\left((\log\log n)^{\log 10}\right)
\end{aligned}
$$

$$
\begin{aligned}
g_6(n) &= \log((\log n)!) \\
&= \Theta(\log n\log\log n) && \blacktriangleleft\ \log(n!) = \Theta(n\log n)
\end{aligned}
$$

Bonus. Ranking Functions 2.0

Extra Slide

- $g_1(n) = \Theta(\log n \log\log n)$
- $g_2(n) = \Theta(n \log\log n)$
- $g_3(n) \approx \Theta\left((\log n)^{\log n + 1/2}\right)$
- $g_4(n) = \Theta(\log\log n)$
- $g_5(n) = \Theta\left((\log\log n)^{\log 10}\right)$
- $g_6(n) = \Theta(\log n \log\log n)$
- $g_7(n) = \Theta(n^3)$
- $g_8(n) = \Theta(2^n)$

Answer: $g_4 < g_5 < g_1 = g_6 < g_2 < g_7 < g_3 < g_8$.

- $g_5 < g_1$: View $g_5(n)$ as $((\log\log n) \cdot (\log\log n)^{2.22\ldots})$. $(\log\log n)^{2.22\ldots} << \log n$.
- $g_7 < g_3$: g_3 is exponential but g_7 is polynomial.
- $g_3 < g_8$: Note that $(\log n)^{\log n} = 2^{\log((\log n)^{\log n})} = 2^{\log n \log\log n}$.
- The rest are clear.